

Redes de Sensores Inalámbricos: Interfaz Web para Shawn

Diego A. Godoy¹, Eduardo O. Sosa^{1,2}, Santiago H. Bareiro¹, Rebeca Díaz Redondo³

¹ Universidad Gastón Dachary, Centro de Investigación en Tecnología de la Información y Comunicaciones, Departamento de Ingeniería y Ciencias de la Producción.

² Secretaría de Investigación y Posgrado (SECIP). Facultad de Ciencias Exactas, Químicas y Naturales, UNaM

³ Departamento de Ingeniería Telemática. E. E. Telecomunicación. Universidad de Vigo
{diegodoy, eduardo.sosa, hbareiro}@citic.ugd.edu.ar,
rebeca@det.uvigo.es

Resumen. En este trabajo se presenta un prototipo de interfaz web para Shawn, un simulador de WSN desarrollado para ser usado desde línea de comandos. Al mismo se le realizaron adaptaciones en su funcionamiento para realizar corridas de simulación y visualización de resultados a través de la Web. Se describen en forma general los componentes del simulador Shawn, la metodología utilizada para el desarrollo del prototipo y finalmente los aspectos fundamentales de la implementación. Conjuntamente se presentan una serie de ensayos para comprobar el funcionamiento de la interfaz Web.

Palabras Claves: Wireless Sensor Networks, Simulación, Shawn.

1 Introducción

El proceso de instalación y configuración de un sistema de simulación de propósito específico, puede ocasionar un consumo importante de tiempo. Ya sea porque requiera ser instalado desde un sistema operativo en particular, o porque es preciso adquirir el hardware y el conocimiento para la instalación correcta del simulador en sí.

Además, luego de realizar la puesta en funcionamiento del simulador, se debe entender cómo proceder con la configuración correcta, para obtener resultados de simulación lo más cercano posible a sistemas reales. Esto implica en algunos simuladores actuales conocer no solo de la problemática a simular, sino también de cómo ingresar los parámetros, dónde ubicar los archivos que son necesarios como fuentes del proyecto y finalmente cómo generar la visualización de los resultados de simulación.

En este sentido, al momento de poner en funcionamiento un sistema de simulación, se buscan herramientas portables o multiplataforma y que tengan una interfaz que abstraiga al usuario de la complejidad necesaria en la configuración del sistema de simulación. Este tipo de herramientas evitan que los tiempos de un proyecto de

simulación se trasladen en mayor medida a la instalación y puesta en marcha del simulador.

Una de las principales características de los sistemas Web es su portabilidad, así también como su facilidad de acceso en forma remota. Con el uso del modelo cliente-servidor, y al utilizar un navegador Web como interfaz, se abstrae al usuario de toda la complejidad interna de un sistema. La complejidad en este caso queda completamente transparente del lado del servidor.

Con la utilización de Shawn [1] como sistema de simulación específico para nodos ISense [2], se ha comprobado que es una herramienta de software poderosa para la simulación de Redes de Sensores Inalámbricos. Pero también se ha observado una gran complejidad para realizar la instalación, configuración y puesta a punto de los módulos del simulador [3]. Estas características y limitaciones hacen que Shawn sea un sistema de simulación especialmente adaptable a la Web.

Si bien ya existen trabajos similares de Simulación Basados en la Web, se considera de gran utilidad realizar una adaptación al ámbito de la simulación de redes, particularmente a las Redes de Sensores Inalámbricos (Wireless Sensor Networks - WSN); donde actualmente se disponen de escasas alternativas que aprovechen la sinergia entre simulación y los sistemas basados en la Web. Entre las alternativas se pueden mencionar a “A Web-Based Integrated Environment for Simulation and Analysis with NS-2”. En el que se desarrolla un completo entorno basado en la web para la simulación y pos-proceso de las salidas generadas por Network Simulation version 2 (NS-2) [4]. La aplicación Web denominada ns2web permite la ejecución remota de simulaciones de redes inalámbrica, incluyendo WSNs y cableadas. También ofrece un conjunto de herramientas para analizar los archivos de rastreo que genera como salida el simulador. Una implementación del simulador se encuentra on-line para público acceso en [5].

Otro ejemplo es “Web-based simulation management: A web-based interface for storing and executing simulation model”, el cual consiste en el desarrollo de un sistema de simulación basado en la web para el lenguaje de simulación SIMAN [6]. El lenguaje SIMAN permite la simulación de sistemas discretos y dinámicos. SIMAN es un simulador de propósito general, lo cual no se condice con las propiedades específicas para realizar simulación de WSN. Para este caso Shawn ofrece más características que se adaptan mejor.

El presente trabajo está estructurado de la siguiente manera: En el apartado 2 se presenta el simulador Shawn, seguidamente, en el apartado 3 se da una introducción a WebML. En el apartado 4 describe el proceso de construcción del simulador y en el apartado 5 se presenta un caso de prueba. Por último en la sección 6 se presentan las conclusiones y trabajos futuros.

2 El Simulador Shawn

Shawn es un framework de simulación cuya idea central es reemplazar los efectos a bajo nivel de una WSN, con modelos abstractos e intercambiables de manera que se pueda utilizar en la simulación de grandes redes en un tiempo razonable [1].

Conceptualmente Shawn se compone de tres partes principales: el Entorno de Simulación, el Secuenciador y los Modelos. El Entorno de Simulación contiene los

elementos simulados y sus propiedades, mientras que el Secuenciador y los Modelos influyen en el comportamiento del Entorno o Ambiente de Simulación [7]. La Figura 1 muestra los principales componentes de Shawn.

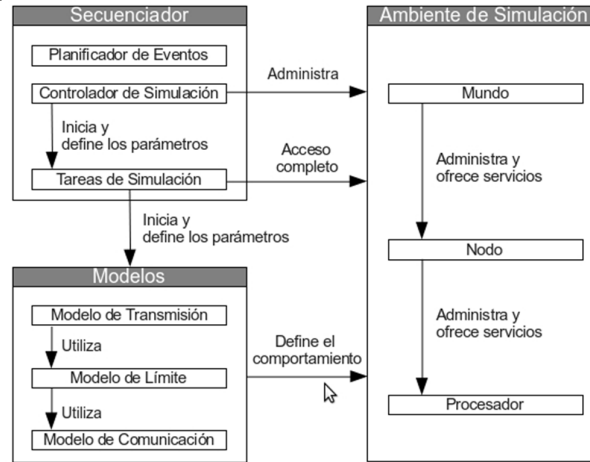


Figura 1. Arquitectura general de los componentes principales de Shawn [7]

La principal característica que hace de Shawn un simulador más rápido [8] que otros simuladores de WSN es su composición interna o diseño. Mientras que otros simuladores se centran en realizar la simulación de un fenómeno en particular, Shawn simula el efecto causado por el fenómeno.

El modelado que se realiza con Shawn de un sistema apunta a la velocidad de simulación a gran escala. En vez de realizar el cálculo de posibles congestiones en una red, analizando cada paquete en forma individual, simplemente se modela la pérdida de paquetes provocados por el alto tráfico.

Este tipo de modelado, se adapta específicamente al paradigma de WSN, donde el foco de la investigación de un problema de software está en entender la estructura fundamental de la red. Una tarea que a menudo está un nivel por encima de los detalles técnicos referente a los nodos individuales y los efectos de bajo nivel.

Un beneficio directo de este tipo de modelado es la escalabilidad. Escenarios visionarios anticipan Redes de Sensores Inalámbricos con un gran número de nodos individuales. Es por ello que un aspecto crítico en el diseño de Shawn es el de poder realizar exitosamente simulaciones de una red con más de 100.000 nodos en un equipo PC estándar [7].

3 WebML

El prototipo del simulador fue desarrollado utilizando Web Modeling Language. WebML es una notación o lenguaje de alto nivel para el modelado, diseño e implementación de aplicaciones Web que hacen uso intensivo de datos [9].

Como se muestra en la Figura 2 el enfoque de WebML para el desarrollo de aplicaciones Web está compuesto por diferentes fases. Éste enfoque está inspirado en el modelo en espiral de Boehm [10] y en línea con los modernos métodos de desarrollo de software. El proceso de desarrollo con WebML se aplica en forma iterativa e incremental, de manera que las fases se repiten y se refinan hasta que los resultados cumplen con los requisitos de determinada aplicación [11]. Este tipo de metodología con ciclos iterativos se adapta especialmente a los desarrollos Web dado la dinámica de cambios que pueden suceder durante el desarrollo y la rapidez con que deben ser desplegados o puestos en funcionamiento los sitios en Internet.

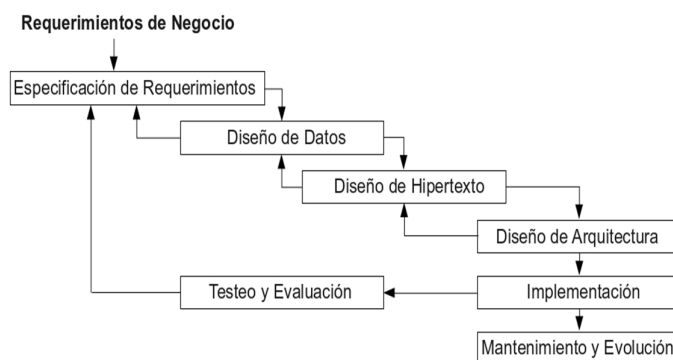


Figura 2. Fases de desarrollo con WebML [12]

A continuación se describe brevemente cada fase [12].

A.- Especificación de Requerimientos: Se centra en la recopilación de información acerca del dominio de la aplicación, las funciones esperadas y sobre la especificación de las funciones por medio de descripciones fáciles de entender.

B.- Diseño de Datos: Esta fase implementa una de las disciplinas más tradicionales y consolidadas en la tecnología de la información. Por lo cual WebML utiliza el modelo de datos entidad-relación o el diagrama de clases de UML.

C.- Diseño de Hipertexto: El modelo de hipertexto permite la definición de la interfaz front-end de la aplicación, la cual se muestra en el navegador del usuario. Esto permite la especificación las páginas y su organización interna en términos de componentes para la visualización del contenido. También se pueden definir los enlaces entre páginas y localización de la información.

D.- Diseño de Arquitectura: Consiste en la especificación del hardware, redes y componentes de software que mejor se adapte a los requerimientos de la aplicación.

E.- Implementación: Es la fase donde se producen los módulos de software necesarios para transformar el diseño de datos y de hipertexto en una aplicación que se ejecute en la arquitectura seleccionada.

F.- Pruebas y Evaluación: Es la actividad de verificación de la conformidad de la aplicación implementada con los requisitos funcionales y no funcionales.

G.- Mantenimiento y Evolución: Abarcan todas las modificaciones efectuadas después que la aplicación ha sido desplegada en un entorno de producción.

H.- Herramientas CASE para modelado con WebML de WebRatio: WebML posee un conjunto de herramientas para modelado asistido por computadora desarrollado por WebRatio [13].

4 Construcción del Simulador

La arquitectura de software para realizar el desarrollo del simulador se apoya en el esquema de Simulación y Visualización Remota [14] y [15]. Donde el motor de simulación y visualización se encuentran completamente implementados en el servidor. El acceso al sistema de simulación es siempre a través del navegador Web. Este último se utiliza como interfaz liviana para ingresar los parámetros de simulación. Los parámetros ingresados son enviados al servidor Web a través de una intranet o bien Internet. El servidor obtiene los parámetros enviados, que luego son dirigidos al motor de simulación Shawn. Una vez que el motor termina completa satisfactoriamente la simulación, los resultados son devueltos al usuario a través de la interfaz web.

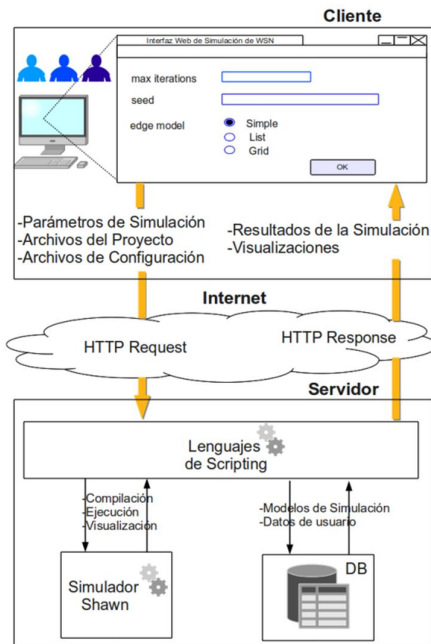


Figura 3 Esquema general del simulador Basado en la Web.

Es importante destacar que, además de ingresar los parámetros que controlan la ejecución de la simulación, también se puede editar a través de la interfaz Web todo el código fuente que dará funcionalidad a los sensores.

De este modo el usuario puede ir guardando sus diferentes modelos de simulación en el servidor para luego realizar pruebas con diferentes soluciones. En la Figura 3 se muestra un esquema general de la solución.

La interfaz de usuario que se ejecuta en el navegador Web del lado del cliente se desarrolla utilizando HTML5, JavaScript, CSS y estará disponible para dispositivos tipo PC. Con el fin de almacenar información relativa a los proyectos de simulación de los usuarios, datos de la aplicación, modelos de simulación y estructuras de datos, se empleará el sistema de gestión de bases de datos PostgreSQL.

El esquema de navegación del simulador utiliza solapas o pestañas que son progresivas en el uso del simulador. En la primera pestaña se puede gestionar todo lo relacionado al árbol de archivos de los diferentes archivos de cada proyecto de simulación. En la Figura 4 se puede apreciar un proyecto con los diferentes archivos que le dan funcionalidad a la simulación.

ShawnWEB Simulación de WSN Basada en la Web

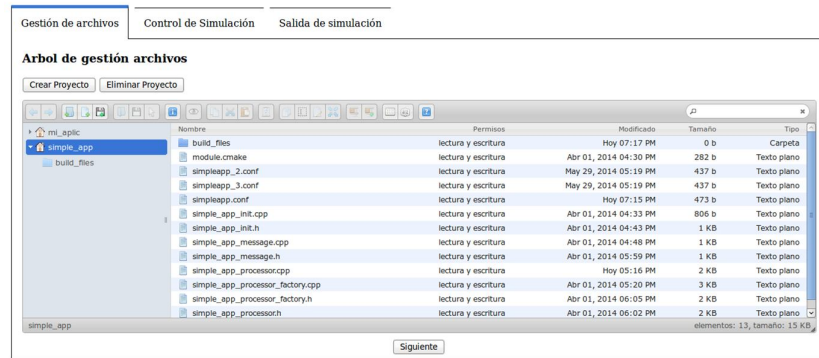


Figura 4. Interfaz Web de gestión de archivos de proyectos de simulación

La siguiente pestaña en la navegación permite ingresar los parámetros de control. En esta página se puede fácilmente controlar alguno de los parámetros más importantes en la simulación. Como se puede ver en la Figura 5, al seleccionar los diferentes archivos de configuración, los parámetros que contienen son cargados en la entrada correspondiente. Una vez cargados en las entradas se pueden modificar y volver a guardar los nuevos parámetros dentro del archivo de configuración seleccionado.

ShawnWEB Simulación de WSN Basada en la Web

Gestión de archivos

Control de Simulación

Salida de simulación

Parámetros de Control

Guardar

Archivo de configuración: simpleapp.conf

count: 22

range: 5

width: 25

height: 25

seed: 1331177

max iterations: 5

Anterior

Siguiente

Figura 5. Interfaz Web de control de simulación

En la última pestaña se puede compilar los archivos fuentes del proyecto, ver las salidas de la compilación y finalmente ejecutar la simulación. Cuando se ejecuta la simulación (siempre y cuando la visualización se especifique en los archivos de control) la visualización generada se puede ver en el navegador Web en diferentes formatos como pdf, png o ps. En la misma pestaña, una vez que obtuvieron los resultados de la simulación, se puede descargar en formato comprimido absolutamente todos los archivos del proyecto que intervienen en la simulación.

ShawnWEB Simulación de WSN Basada en la Web

Gestión de archivos

Control de Simulación

Salida de simulación

Salida de compilación

Salida de ejecución

Compilar

Limpiar Salida

-- Configuring done
-- Generating done
-- Build files have been written to: /var/www/prueba/shawn/buildfiles
LA OPCION SHAWN.WEB ESTA ACTIVADA
NOMBRE DE LA APLICACION A COMPILAR: simple_app
Scanning dependencies of target shawn_web_simple_app
[20%] Building CXX object CMakeFiles/shawn_web_simple_app.dir/legacyapps/simple_app/simple_app_message.o
[40%] Building CXX object CMakeFiles/shawn_web_simple_app.dir/legacyapps/simple_app/simple_app_processor.o
[60%] Building CXX object CMakeFiles/shawn_web_simple_app.dir/legacyapps/simple_app/simple_app_processor_factory.o
[80%] Building CXX object CMakeFiles/shawn_web_simple_app.dir/legacyapps/simple_app/simple_app_init.o
[100%] Building CXX object CMakeFiles/shawn_web_simple_app.dir/legacyapps/shawn_web_simple_app
Linking CXX executable /var/www/prueba/shawn/src/legacyapps/simple_app/build_files/shawn_web_simple_app
[100%] Built target shawn_web_simple_app

Ejecutar

Descargar

Visualizar

processors.simple_app: Nodo ID: v19-Z4uRB-V Recibió un mensaje de 'v13-Z4uRB-P'
processors.simple_app: Nodo ID: v3-Z4uRB-F Recibió un mensaje de 'v14-Z4uRB-Q'
processors.simple_app: Nodo ID: v2-Z4uRB-E Recibió un mensaje de 'v15-Z4uRB-R'
processors.simple_app: Nodo ID: v9-Z4uRB-L Recibió un mensaje de 'v15-Z4uRB-R'
processors.simple_app: Nodo ID: v10-Z4uRB-M Recibió un mensaje de 'v15-Z4uRB-R'
processors.simple_app: Nodo ID: v2-Z4uRB-E Recibió un mensaje de 'v17-Z4uRB-T'
processors.simple_app: Nodo ID: v10-Z4uRB-M Recibió un mensaje de 'v17-Z4uRB-T'
processors.simple_app: Nodo ID: v4-Z4uRB-G Recibió un mensaje de 'v18-Z4uRB-U'
processors.simple_app: Nodo ID: v9-Z4uRB-L Recibió un mensaje de 'v18-Z4uRB-U'
processors.simple_app: Nodo ID: v20-Z4uRB-W Recibió un mensaje de 'v18-Z4uRB-U'
processors.simple_app: Nodo ID: v0-Z4uRB-C Recibió un mensaje de 'v19-Z4uRB-V'
processors.simple_app: Nodo ID: v11-Z4uRB-N Recibió un mensaje de 'v19-Z4uRB-V'
processors.simple_app: Nodo ID: v13-Z4uRB-P Recibió un mensaje de 'v19-Z4uRB-V'
processors.simple_app: Nodo ID: v4-Z4uRB-G Recibió un mensaje de 'v20-Z4uRB-W'
processors.simple_app: Nodo ID: v9-Z4uRB-L Recibió un mensaje de 'v20-Z4uRB-W'
processors.simple_app: Nodo ID: v18-Z4uRB-U Recibió un mensaje de 'v20-Z4uRB-W'
..... DONE ITERATION 1
[22 active, 0 sleeping, 0 inactive]
..... BEGIN ITERATION 2
..... DONE ITERATION 2

Anterior

Figura 6. Interfaz Web de salida de simulación

5 Pruebas

Para realizar las pruebas se desarrolló una aplicación denominada SimpleApp. Como su nombre indica, es una aplicación bastante simple. Su objetivo es simplemente para probar el comportamiento del simulador y la interfaz Web. El caso de prueba se describe a continuación: En la primera ronda de simulación, cada nodo envía un único mensaje. El nodo que recibe el mensaje imprime su propio

identificador dentro de la red, así como el identificador del remitente. Lo que se puede observar con esta aplicación es principalmente la conectividad de los nodos.

Para simplificar en este caso solamente se ingresaron como parámetros a través de la interfaz Web seis parámetros; la cantidad de nodos (count), el rango de cobertura del sensor (range), las medidas de ancho y el alto del medio rectangular (width y height), la semilla (seed) de la simulación, y cantidad de iteraciones (max iterations).

En la Tabla 1 se puede visualizar los parámetros ingresados en la primera prueba.

Tabla 1. Parámetros de control ingresados en la primera prueba de Simulación

Parámetro	Valor
count	22
range	1
width	7
height	7
seed	1331177
max iterations	5

Luego de correr la simulación, mediante las visualizaciones generadas (Figura 7) que algunos nodos no tienen conexión y quedan sin comunicación con los otros nodos de la red de sensores.

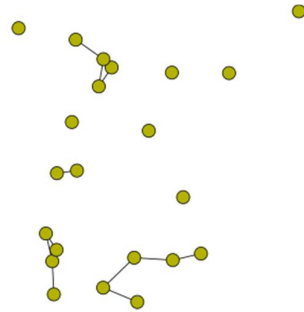


Figura 7. Visualización generada por ShawnWEB de interconexión con algunos nodos sin conexión

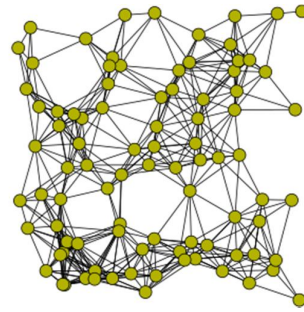


Figura 8. Visualización generada por ShawnWEB de interconexión entre todos los nodos

Modificando la cantidad de nodos y el rango de cobertura, como se puede visualizar en la **Tabla 2** se puede observar que todos los nodos quedan totalmente interconectados (Figura 8).

Mediante estas pruebas se pudo constatar las ventajas que otorga el prototipo desarrollado. Permite realizar en forma ágil modificaciones tanto en los parámetros de control como archivos fuentes de simulación. Además, luego de hacer las modificaciones, se puede fácilmente obtener visualizaciones para comprobar los efectos generados sobre la red.

Tabla 2. Parámetros de control ingresados en la segunda prueba

Parámetro	Valor
count	90
range	1.5
width	7
height	7
seed	1331177
max iterations	5

6 Conclusiones y Trabajos Futuros

En el presente trabajo se ha avanzado en la implementación de un prototipo de simulador basado en Web utilizando como motor de simulación el simulador Shawn. El desarrollo final del simulador servirá en un futuro para que investigadores y estudiantes puedan realizar simulaciones de WSN enfocándose solamente en la aplicación y el diseño de la red, sin necesidad de tener que lidiar con cuestiones de instalación y de funcionamiento de Shawn. Actualmente se utiliza en simulaciones para diversos proyectos de investigación relacionados con ciudades inteligentes como: a) Redes Inalámbricas de Sensores: Una experiencia en la Industria del Té, cuyo objetivo es realizar una experiencia en una empresa productora de Té negro de la provincia de Misiones interesada en mantener mejor control en las variables del proceso utilizando una tecnología que es clave en Internet del Futuro; b) Análisis Y Comparación De Modelos De Propagación Para Optimizar La Localización Geográfica Del Ganado, cuyo objetivo es comparar modelos de propagación de señal de radiofrecuencia con el fin de optimizar la localización geográfica del ganado en un escenario de terrenos con topografía irregular y diferentes estratos de vegetación.

Como trabajos futuros se propone la incorporación de más opciones de seguridad en el sistema Web, la posibilidad de facilitar la edición de los archivos xml que determinan los límites del ambiente. También se pueden implementar nuevas formas modificar online los archivos fuentes de los distintos proyectos de simulación.

7 Bibliografía

- [1] Stefan Fischer, Dennis Pfisterer, and Sándor P. Fekete, “*Shawn: The fast, highly customizable sensor network simulator*”. Braunschweig University of Technology and University of Lubeck, Alemania, 2007. ISBN 1-4244-1231-5.
- [2] Coalesense. (2013) iSense Wireless Sensor Network Software. <http://www.coalesenses.com/index.php?page=isense-software>
- [3] Shawn Community. (2013) Wiki de Shawn. <https://github.com/itm/shawn/wiki>
- [4] NS2 Community. (2011) NS2 Wiki. [Online]. <http://nsnam.isi.edu/nsnam>
- [5] NS2 WEB Community. (2013) NS2 WEB.

[<http://vlssit.iitkgp.ernet.in/ns2web/ns2web>

- [6] C. Dennis Pegden, *"Introduction to SIMAN"*. Proceedings of the 15th conference on Winter simulation-Volume 1. IEEE Press, 1983.
- [7] A. Kroeller, D. Pfisterer, C. Buschmann, and S.P. Fekete, *"Shawn: A new approach to simulating wireless sensor networks"*. Alemania, 2005.
- [8] Shawn. (2013, June) Shawn: The fast, highly customizable sensor network simulator. <http://goo.gl/kCe9I3>
- [9] Stefano Ceri, Piero Fraternali, and Aldo Bongio, *"Web Modeling Language (WebML): a modeling language for designing Web sites"*. Computer Networks, 2000.
- [10] B. Boehm, *"A Spiral Model of Software Development and Enhancement"*. IEEE Computer, 1988.
- [11] Marco Brambilla, Sara Comai, Piero Fraternali, and Maristella Matera, *"Designing web applications with WebML and WebRatio"*. Web engineering: Springer, Londres, 2008.
- [12] Stefano Ceri et al., *"Designing data-intensive Web applications"*:. Morgan Kaufmann, 2003. ISBN 1-5586-0843-5.
- [13] Web-Ratio. (2014) Web-Ratio: La Nueva Ecuación de los Negocios-TI. <http://www.webratio.com/>
- [14] Larry Whitman, Brian Huff, and Senthil Palaniswamy, *"Commercial simulation over the web"*. 30th conference on Winter simulation: IEEE Computer Society Press, 1998.
- [15] James Byrnea, Cathal Heaveya Heaveya, and P.J. Byrne, *"A review of Web-based simulation and supporting tools"*. Simulation Modelling Practice and Theory Vol 18: Elsevier Publisher, 2010.